

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-086373

(43)Date of publication of application : 26.03.2002

(51)Int.CI.

B25J 5/00

B25J 13/08

(21)Application number : 2000-275690

(71)Applicant : NATIONAL INSTITUTE OF
ADVANCED INDUSTRIAL &
TECHNOLOGY
TAKEUCHI HIROYOSHI

(22)Date of filing : 11.09.2000

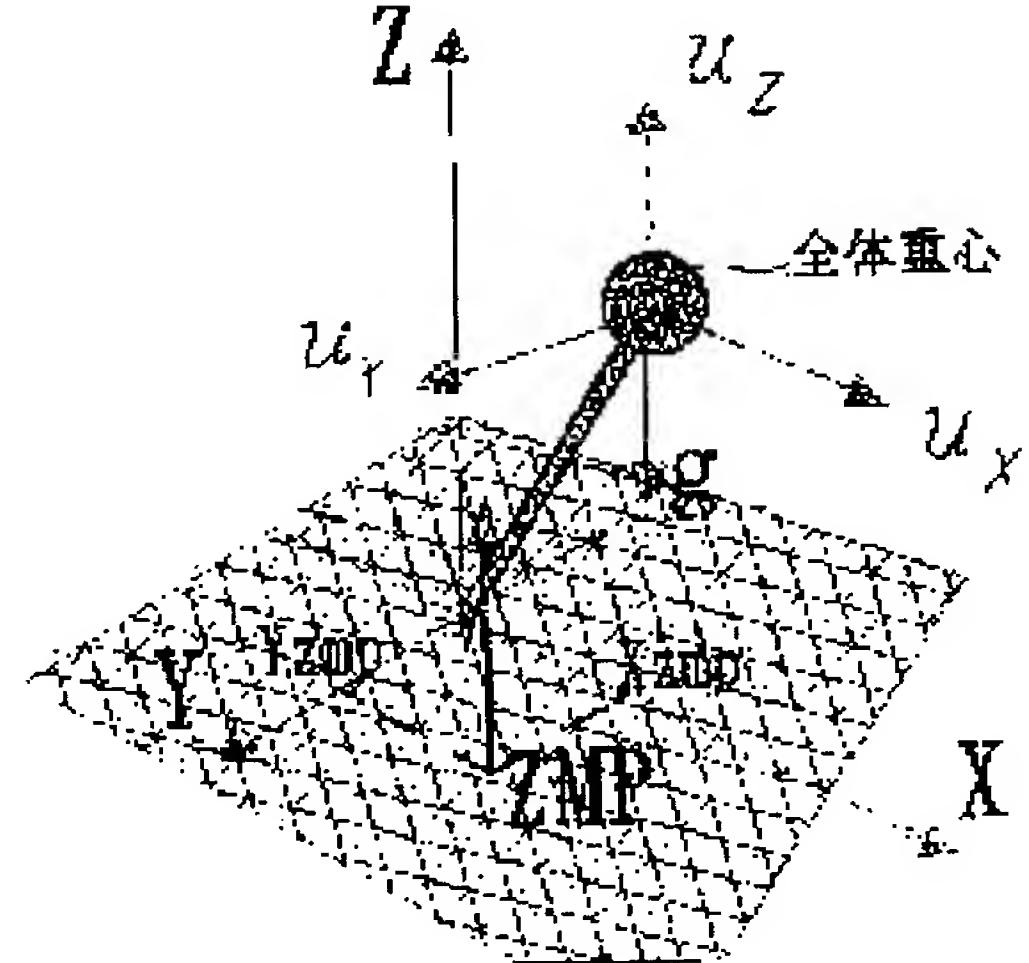
(72)Inventor : TAKEUCHI HIROYOSHI

(54) REAL-TIME OPTIMAL CONTROL METHOD FOR LEG TYPE ROBOT

(57)Abstract:

PROBLEM TO BE SOLVED: To provide zero moment point control using RHC(receding horizon control) capable of real-time optimizing calculation for a leg type robot, and control the robot through combined use of trajectory and non-linear controls using a zero moment point automatically generated in real time.

SOLUTION: In the real-time optimal control method for the leg type robot, an optimal target ZMP(zero moment point) trajectory is obtained in real time when the zero moment point of the leg type robot satisfies an equality constraint of an expression: $xzmp(uz-g)=x(uz-g)-zux$, and when an optimal solution is obtained by use of an expression: $uzmp(uz-g)=x(uz-g)-zux$ with $xzmp$ used as new input $uzmp$.

 $\pi_{mp}(x, -g) = x(u_z - g) - zux$ $\pi_{mp}(u_z - g) = x(p_z - g) - zv_x$

(1.2)

(1.3)

LEGAL STATUS

[Date of request for examination] 11.09.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3677623

[Date of registration] 20.05.2005

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(2)

1

【特許請求の範囲】

【請求項1】脚式ロボットのZMPが

$$x_{zmp}(u_z - g) = x(u_z - g) - zu_x \quad (15)$$

の等式拘束条件を満足するものにおいて、 x_{zmp} を新たな入力 u_{zmp} として

$$u_{zmp}(u_z - g) = x(u_z - g) - zu_x \quad (16)$$

を用いて最適解を求めたときに最適な目標ZMP軌道をリアルタイムで得るようにしたことを特徴とする脚式ロボットのリアルタイム最適制御方法。

【請求項2】ロボット全体重心の目標軌道と、請求項1の最適な目標ZMP軌道生成とともにロボットの各関節の制御を非線形制御により行うことを特徴とする脚式ロボットのリアルタイム最適制御方法

【請求項3】請求項1の方法に基づいて、予め、各パラメータを変化させた場合の計算結果を集計し、設定したロボット動作に対する目標ZMP軌道の足裏内の動作範囲を把握可能とすることによって、ロボットの足裏面積を適切に設定することが可能となることを特徴とする脚式ロボットのリアルタイム最適制御方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、リアルタイム最適化計算が可能なリシーディング・ホライゾン制御 (Receiving Horizon Control)。以下「R★

$$x_{zmp} = \frac{\sum_{i=0}^n m_i(-g + \ddot{z}_i)x - \sum_{i=0}^n m_i \ddot{x}_i z_i}{\sum_{i=0}^n m_i(-g + \ddot{z}_i)x} \quad (1)$$

と表すことが出来る。ここでロボット全体の重心位置で代表すれば上式は簡単に

【数2】

$$x_{zmp} = \frac{(-g + \ddot{z})x - \ddot{z}\ddot{x}}{-g + \ddot{z}} \quad (2)$$

とかける。即ち、ロボットの原点まわりの総モーメントが、ZMPと床反力から得られるモーメントと釣合う点である。ZMPが、重心と同じように、足裏平面によって構成される多角形内に収まっているれば、ロボットは安定して動的歩行をことができ、外れるとロボットは

☆転倒を起こし、歩行が継続できない。

【0004】従来の脚式ロボットにおいて問題であったのは、(2)式でZMPを算出しようとしたときに、 x と z の2変数があるために解が一意に決定しないことがある。従来これを回避するために、例えば重心位置を一定高度 h_{ref} に保つ条件を仮定すると

【数22】

$$\ddot{z} = 0$$

となり、

【数3】

$$x_{zmp} = \frac{gx - h_{ref}\ddot{x}}{g} \quad (3)$$

とすることにより、擬似的なZMPを算出するような方法を取らざるを得なかった。しかし、これはあくまでも擬似的な解が得られるだけで、真のZMPは得られな

◆い。そして例えば、重心加速度を、

【数4】

$$\ddot{x} = \frac{1}{h_{ref}}(gx - K(\text{目標ZMP} - \text{実全床反力中心点})) \quad (4)$$

のように入力することができれば実全床反力が目標ZMPに収束するような復元トルクが得られる。

【0005】しかし、このように重心高度を一定に保つという点において、設計者の意思が入り込んでしまっているが、これは、(4)式で高度一定にするような手法をとらないとZMPが一意に定まらないからである。真的ZMPを従来の方法で求めるには、ZMPを予め最適化問題として定式化し、勾配法等を用いて計算する必要がある。勾配法による最適化の場合、繰り返し計算を必要とするので、このような最適化計算は通常オフライン計算となる。しかし、ロボット制御のようなリアルタイム性が要求される計算では、オフライン最適化計算は予期しない障害物が現れたような場合、対処できないという問題があった。

*

$$x_{zmp}(u_z - g) = x(u_z - g) - zu_z$$

(15)

の等式拘束条件を満足するものにおいて、 x_{zmp} を新たに入力 u_{zmp} として

※

$$u_{zmp}(u_z - g) = x(u_z - g) - zu_z$$

(16)

を用いて最適解を求めたときに最適な目標ZMP軌道をリアルタイムで得るようにした脚式ロボットのリアルタイム最適制御方法（以下、単に「第一の発明」ともいう。）を特徴とする。本発明は、ロボット全体重心の目標軌道と、上記第一の発明に基づく最適な目標ZMP軌道生成とともにロボットの各関節の制御を非線形制御により行うようにした脚式ロボットのリアルタイム最適制御方法を特徴とする。即ち本発明では、ロボットがリアルタイム制御されるために必要なすべての目標軌道がリアルタイムに生成され、通常の非線形制御によって実際のロボットが制御される。また、RHC定式化において、ロボットの非線形性も含めて定式化すれば、ロボットのすべてのリアルタイム制御自体もRHCのみによって可能となる。また、本発明は、上記第一の発明に基づく★

$$\dot{x} = f(t, x(t), u(t))$$

とする。fは、必要なだけの微分可能性を有する関数とする。等式拘束条件として、

$$C(x(t), u(t), t) = 0$$

とする。このときのハミルトンHは、

$$H = L + \lambda^T f + \rho^T C$$

となる。Lはラグランジアン、 λ と ρ は未定乗数である。

【0010】Euler-Lagrange方程式（オイラー、ラグランジュ方程式）は、

【数8】

*【0006】

【発明が解決しようとする課題】本発明は、リアルタイム最適化計算が可能なRHCを用いた、脚式ロボットのZMP制御を提供することを目的とする。また、本発明は、リアルタイムで自動生成される目標ZMP軌道と、非線形制御と併用してロボットを制御することを特徴とする。

【0007】

【課題を解決するための手段】本発明は、上記の課題を解決することを目的として、脚式ロボットのZMPが【数15】

*

※【数16】

※

★いて、予め、各パラメータを変化させた場合の計算結果を集計し、設定したロボット動作に対する目標ZMP軌道の足裏内の動作範囲を把握可能とすることによって、ロボットの足裏面積を適切に設定することが可能となることを特徴とする脚式ロボットのリアルタイム最適制御方法を特徴とする。

【0008】

【発明の実施の形態】ここで、等式拘束を含んだRHCについて説明する。従来、RHCの定式化において、等式拘束条件を含む定式化は行われていなかった。

【0009】xを状態変数、uを制御入力、tを時間とし、RHCのリアルタイム制御の制御対象の状態方程式を、

【数5】

(5)

☆【数6】

☆

◆◆【数7】

(6)

(7)

【数8】

$$\dot{\lambda} = -H_x^T$$

$$H_u = 0$$

$$\dot{x} = f$$

(8)

$$C = 0$$

$$x[x(t), 0] = 0$$

$$\psi[x(t), T] = 0$$

10

*

$$\delta x = f_x \delta x + f_u \delta u + f_p \delta p$$

$$\dot{\delta \lambda} = -H_{xx} \delta x - f_x^T \delta \lambda - H_{uu} \delta u - H_{up} \delta p \quad (9)$$

$$H_{ux} \delta x + f_u^T \delta \lambda + H_{uu} \delta u + H_{up} \delta p = 0$$

$$C_u \delta u + C_x \delta x = 0$$

として考えられる。ここで、 $H_u = 0$ 、 $H_p = 0$ (i.e. $C = 0$) の微少変化の2式を連立方程式として解くと δu 、 δp が消去できる。

※20

$$\frac{d}{dt} \begin{bmatrix} \delta x \\ \delta \lambda \end{bmatrix} = \begin{bmatrix} A & -B \\ -C & -A^T \end{bmatrix} \begin{bmatrix} \delta x \\ \delta \lambda \end{bmatrix} \quad (10)$$

を得る。但し、

★ ★ 【数11】

$$A = f_x - f_u H_{uu}^{-1} H_{ux}$$

$$B = f_u H_{uu}^{-1} f_u^T \quad (11)$$

$$C = H_{xx} - H_{uu} H_{uu}^{-1} H_{ux}$$

である。これで等式拘束条件を含有した時変線形方程式が得られた。後は、連続変形法 (Continuation Method) に従って、リアルタイム最適解が得られる。この点については、参考文献 (大塚敏之著、「非線形最適フィードバック制御のための実時間最適化手法」、計測と制御 vol. 36 no. 11 p. 776-783 1997) 等を参照されたい。

【0013】ここでは、2脚式ロボットや4脚式ロボットをモデル化するときに最も基本的な例として、ロボット全体を1つの代表質点として扱う。ロボット全体を1質点の倒立振子として扱うことは従来から一般に行われている手法である。このような代表質点によるモデル化によれば、後述するように、2脚式ロボットや4脚式ロボット、その他の多脚式ロボットに簡単に応用することができる。

【0014】そして、系の入力として、X、Z方向の加速度を設定する。即ち、図3において

【数12】

$$u_x = \ddot{x}, u_z = \ddot{z} \quad (12)$$

とおく。鉛直下方には重力が加わる。従って、状態方程式として、

【数13】

$$\frac{d}{dt} \begin{bmatrix} x \\ z \\ \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{z} \\ u_x \\ u_z - g \end{bmatrix} \quad (13)$$

とする。これは矢状面 (Sagittal 平面) 上で考
えられているが、正面 (Frontal 平面) で考えれ
【数 14】

$$\frac{d}{dt} \begin{bmatrix} y \\ z \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \dot{y} \\ \dot{z} \\ u_y \\ u_z - g \end{bmatrix} \quad (14)$$

となる。

【0015】次に等式拘束条件を考えると、脚式ロボットは、各時刻において、ZMP は以下のような等式拘束条件を満足していかなければならない。ZMP は、総モーメントが 0 となる点である。これが、足裏または接地脚先によって構成される多角形内にあれば、モーメントを発生せずに床反力を支持することができる。

【数 15】

$$x_{zmp}(u_z - g) = x(u_z - g) - zu_x \quad (15)$$

なお、 g は重力の加速度である。

【0016】これは、図 3 のように、質点に加わる加速度入力や、重力による総モーメントが釣り合うことを意味している。ZMP は、これが足裏内または接地脚先によって構成される多角形内に位置すれば、モーメントが発生せずに床反力を支持することができる。しかし、最適制御問題の定式化の中で、ZMP を状態変数の 1 つと★

★して状態方程式内に含めようすると ZMP の微分をとらなくてはならなくなり、工夫を要する。そこで、ここでは、式 (15) において、 x_{zmp} を新たな入力 u_{zmp} として考えることにより、うまく定式化できるこ
とを示す。

【0017】従って、式 (15) は次のようになる。

【数 16】

$$u_{zmp}(u_z - g) = x(u_z - g) - zu_x \quad (16)$$

このように、ZMP を入力の一つとして扱えば、最適解を求めたときに最適な ZMP 入力も同時に得られることになる。この点において、等式拘束条件は重要な役割を持っている。

【0018】評価関数は、入力のノルムとする。ここで
は、 u_{zmp} を評価関数内に含めている点が特徴であ
る。このように設定することにより、各軸加速度と ZMP のノルムを最小に抑える解を求めるこ

とになる。ZMP を、足裏内でなるべく揺動しないように設定することはロボットの足裏をなるべく小さく設計できることにつながる。

【数 17】

$$J = X S_f X^T + \int_t^{t+T} [u_x(t, \tau) \ u_z(t, \tau) \ u_{zmp}(t, \tau)] R \begin{bmatrix} u_x(t, \tau) \\ u_z(t, \tau) \\ u_{zmp}(t, \tau) \end{bmatrix} d\tau \quad (17)$$

$$X = [x_f - x(t_f) \ y_f - y(t_f) \ \dot{x}_f - \dot{x}(t_f) \ \dot{y}_f - \dot{y}(t_f)]$$

また、ZMP 入力 u_{zmp} 以外の項を異なる数式によつて構成した評価関数も考えられる。即ち、本発明の範囲

は、 u_{zmp} が含まれた評価関数すべてについて請求している。具体的には、式 (17) 上段の式の「L」部に

において、L内に、 $u^2_{z_{\text{mp}}}$ が含まれる式Jのすべてについて請求している。評価関数はRHCの場合、時刻tからt+Tまでの区間が移動していくことになる。

【0019】数値計算の実際の例を図4に示す。使用したパラメータは、質量m=1.0kg、始点、及び終点における水平方向速度0.5m/sec、鉛直方向速度0.0m/secとした。初期値は、(x, z) = (-0.25, 0.8, 0.5, 0.0) 目標値は(x, z) = (0.25, 0.8, 0.5, 0.0) である。また、R=(1.0, 1.0, 1.0), Sf=(1.0, 440.0, 1.0, 0.0)である。図4(a)の目標値に移動してい*10

$$u_{z_{\text{mp}}}(u_z - g) = x(u_z - g) - zu_x \quad (16)$$

の左辺と右辺を時間履歴に沿って各々計算したものである。図より左辺と右辺はほぼ均衡が保たれており、逐次生成されるZMP値が妥当なものであることがわかる。この計算に要したCPU時間は、0.09(sec)であり、シミュレーション時間0.5(sec)に対して充分短い時間で完了でき、リアルタイム制御が可能であることがわかる。

【0021】このように本実施例で示した方法によれば、リアルタイムにZMP入力を生成できるため、予め20 ZMP軌道をオフライン計算で求めておくような従来の※

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ u_x \\ u_y \\ u_z - g \end{bmatrix} \quad (18)$$

拘束条件は、図3に示すように、x軸方向の釣合とy軸方向の釣合を考慮しなければならない。★

$$x_{\text{mp}}(u_x - g) = -u_x z + (u_z - g)x \quad (19)$$

$$y_{\text{mp}}(u_z - g) = -u_y z + (u_z - g)y$$

拘束条件の数を減らすためには、2式をまとめて、★★【数20】

$$u_x z (y_{\text{mp}} - y) = u_y z (x_{\text{mp}} - x) \quad (20)$$

とすることも可能である。

◆ルムを最小にする。

【0023】評価関数は、各軸加速度とZMP入力のノム40 【数21】

$$\begin{aligned} J &= X S_f X^T + \int_t^{t+T} U R U^T d\tau \\ U &= [u_x(t, \tau) \ u_y(t, \tau) \ u_z(t, \tau) \ u_{x_{\text{mp}}}(t, \tau) \ u_{y_{\text{mp}}}(t, \tau)] \\ X &= [x_f - x(t_f) \ y_f - y(t_f) \ z_f - z(t_f) \ , \\ &\quad \dot{x}_f - \dot{x}(t_f) \ \dot{y}_f - \dot{y}(t_f) \ \dot{z}_f - \dot{z}(t_f)] \end{aligned} \quad (21)$$

ここでも、本発明の範囲は、2次元の場合と同様に、式(21)最上段の式の「L」部において、L内に、 $u^2_{x_{\text{mp}}} + u^2_{y_{\text{mp}}}$ が含まれる式Jのすべてについて請求している。以上の定式化を踏まえてシミュレー

*くに伴い、図4(g)でZMPも移動している様子が確認できる。この例では、系の挙動を見易いように、高度0.8mに対し、水平方向位置は1.0secで0.5m進む。脚式ロボットとしては少し大きな挙動となっている。本発明の実施例では、このような大きな挙動でもZMPは生成される。

【0020】図4(i), (j)は、生成されたZMP値が妥当な値か確認するため、拘束条件式、【数16】

※手法に比較して非常に便利である。図4(h)は、RHCを計算するまでの誤差Fの値の推移である。このシミュレーションでは、F値は途中で少し大きくなるものの、全体的には充分低い値に抑えられている。

【0022】ここまで説明を分かり易くするために2次元平面で述べたが、この応用として3次元空間での定式化も可能である。3次元質点の状態方程式として、【数18】

11

ション計算を行った結果を図5(a)～図6(o)に示す。初期値は、(-0.125,-0.125,0.800,0.500,0.500,0.000)、目標値は、(0.125,0.125,0.800,0.500,0.500,0.000)である。また、 $R = (1.0, 1.0, 1.0, 1.0, 1.0, 1.0)$ 、 $S_f = (100.0, 600.0, 3400.0, 0.1, 2.0, 0.1)$ である。(j) (k)は各軸のZMP入力の推移である。図より、x軸上では、±0.1m以内、y軸上では、±0.05m以内に収まっている様子がわかる。この範囲であれば、ロボット脚の妥当な足裏を設計することができる。また、(l) (m)は、20式の等式拘束条件の左辺値と右辺値を示しており、ほぼ拘束条件を満たしている。

【0024】本発明を実施する方法は、2通りの方法が考えられる。第一は、RHCと非線形制御の併用である。ロボット本体は一般に非線形な機械的リンク系であり、実機への応用に当たっては上述のリアルタイムにZMPを生成する方法と、従来の非線形制御とを併用することが考えられる。即ち、ロボットの目標軌道は、上述の方法によりリアルタイムに生成し、ロボットの各関節の制御は現代制御理論の非線形制御により行う。従来の脚式ロボットでは、目標軌道はオフライン計算によって予め求めておいた各軌道をいくつか用意しておく必要があった。しかし、これでは例えば障害物を回避しながら脚を進めるといった行動が不可能である。この点において、本実施例の方法は有用性がある。

【0025】実機では、現在のZMP値を力センサ等にて測定することが可能であるため、RHCにて逐次得られる u_{ZMP} 値を新たに目標値と考えて、実際のZMPがこれに追従するように制御系を組む。 u_x 、 u_z についても加速度入力であるため、これを新たに加速度目標値と考えて、実機の重心がこの加速度に追従するような制御系とすればよい。このように、本実施例の方法は、ロボット全体重心についてモデル化したものなので、2脚式ロボットや4脚式ロボットあるいはそれ以外の脚式ロボットにも応用が可能である点が利点である。第2は、すべての処理をRHCで行う方法である。RHCの定式化の際、状態方程式を全体重心で代表するのではなく、ロボットの各リンクの非線形性を考慮した非線形状態方程式をたてる。この方法は、厳密なモデルによって非線形性が考慮されるが、脚の数が変更されたりロボットの構造が違うと、モデル化を初めからやり直さなければならない。

【0026】幅や歩行速度が決定されると、遊脚の振り出し等で生じる非線形力は、図7の非線形制御部において行う。ロボットを設計する際に、前記第一の発明に基づいて、予め、各パラメータを変化させた場合の計算結

10

12

果を集計しておき、設定したロボット動作に対する目標ZMP軌道の足裏内の動作範囲を把握可能とすることによって、ロボットの足裏面積を適切に設定することが可能となる。

【0027】4脚式の場合、2脚づつを交互に支持脚、遊脚として扱うことができる。2脚先端が構成する平面は、2脚式ロボットの場合に比較して、広い面積を確保することができる。例えば4脚式ロボットがTrrot歩容を行う場合、対角どおしの脚を交互に支持脚期-遊脚期を繰返す必要がある。このときには、図9のように、平面内の始点から、面内を横切り、次の支持脚ペアの平面に突き当たる点を終端と設定すれば、ZMPはその周辺を移動する。ZMPが収まるように平面面積をとっておけば、安定した歩行パターンが実現される。

【0028】

【発明の効果】本発明に係る脚式ロボットのリアルタイム最適制御方法においては、リアルタイム最適化計算が可能なRHC制御(Receding Horizon Control)を用いた、脚式ロボットのZMP(Zero Moment Point)制御を得ることができる。この結果、ロボットの目標軌道をリアルタイムで生成することができ、従来のように、目標軌道をオフライン計算によって予め求めておく必要がなく、したがって、例えば障害物を回避しながら脚を進めるといった臨機応変な行動を可能とするものである。

【図面の簡単な説明】

【図1】脚式ロボットのZMPを示す説明図である。

【図2】脚式ロボットの矢状面(sagittal平面)と正面(frontal平面)を示した図面である。

【図3】3次元空間における動的釣合いを示した図である。

【図4】数値計算した実際の例を示した図である。

【図5】3次元空間において定式化した場合のシミュレーション結果を一部示した図である。

【図6】図5に統いて3次元空間において定式化した場合のシミュレーション結果の残りを示した図である。

【図7】制御ブロック線図である。

【図8】2脚式ロボットの歩行状態を示した説明図である。

【図9】4脚式ロボットの歩行状態を示した説明図である。

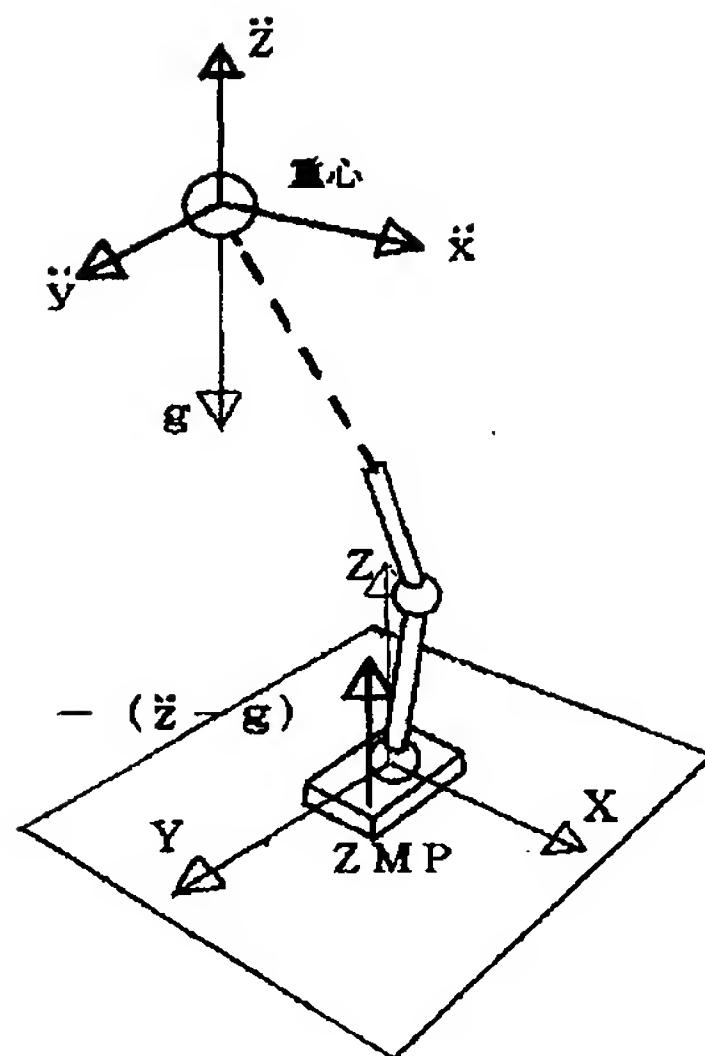
【符号の説明】

ZMP ゼロ・モーメント・ポイント

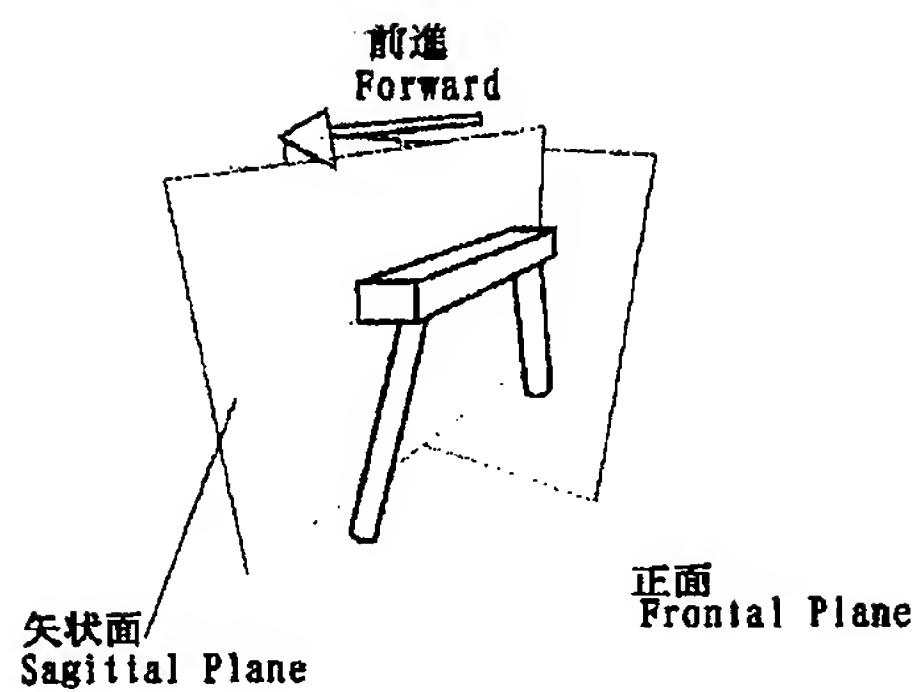
30

40

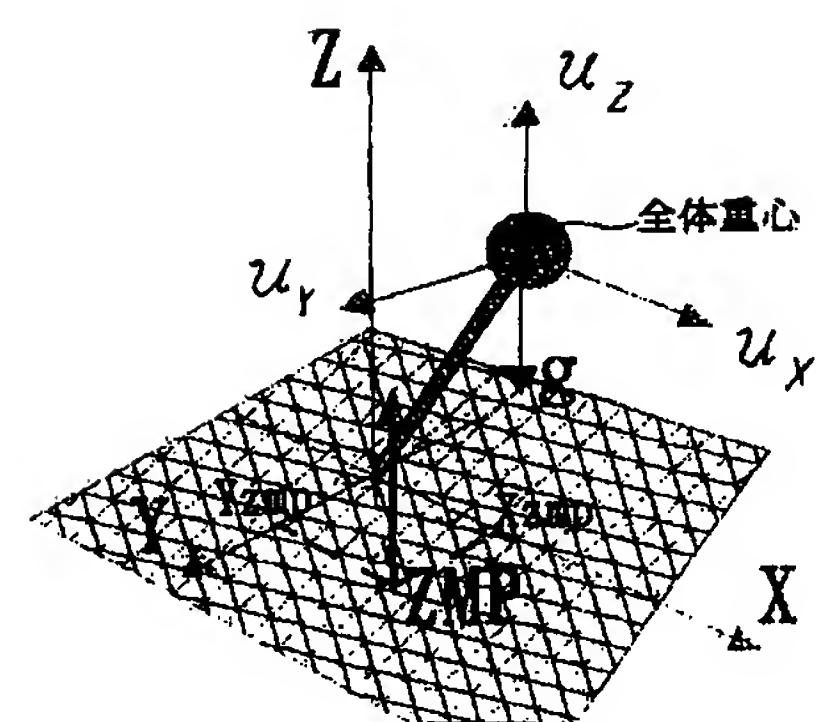
【図1】



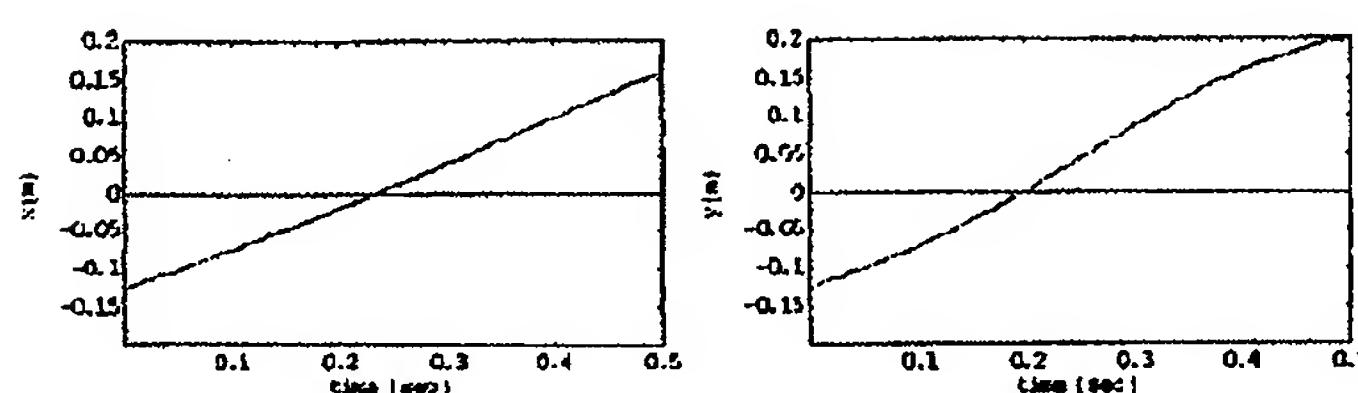
【図2】



【図3】

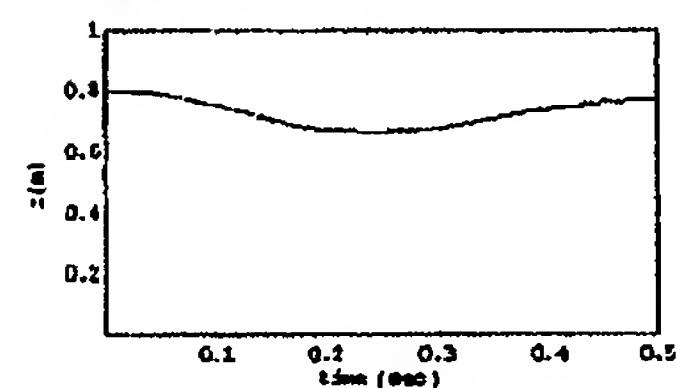


【図5】

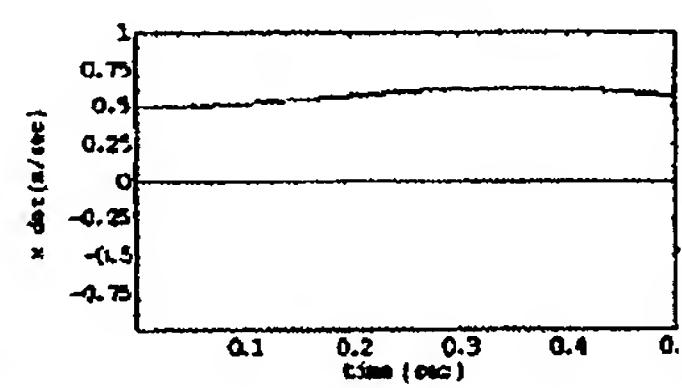


(a) trajectory of X

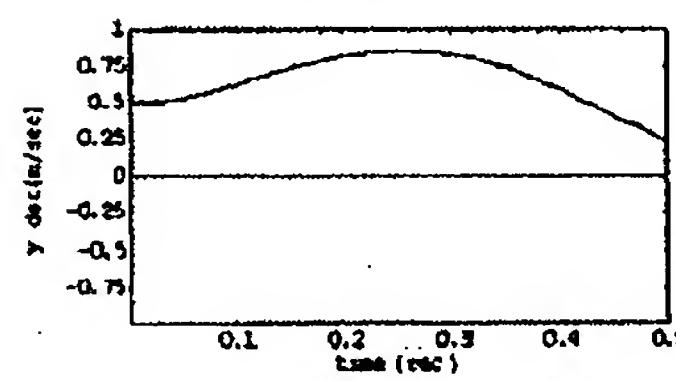
(b) trajectory of Y



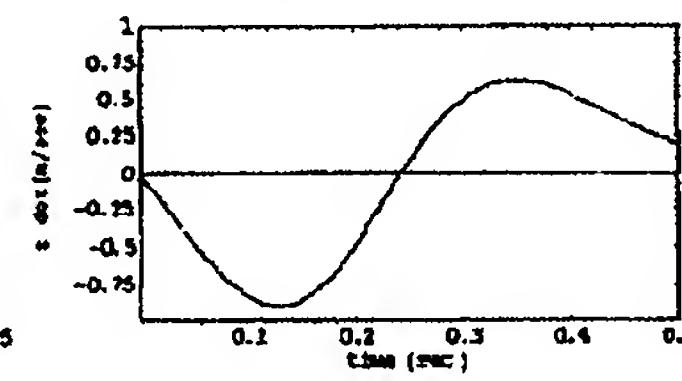
(c) trajectory of Z



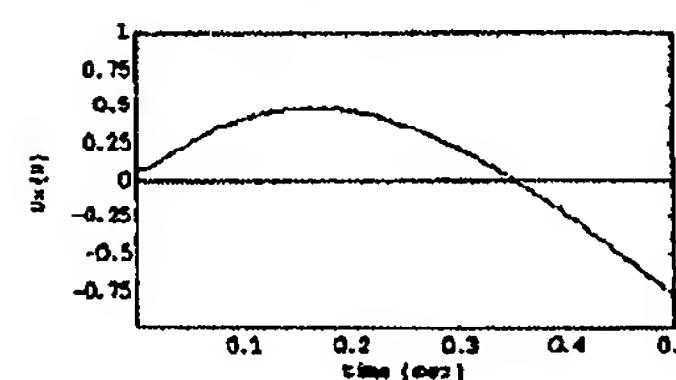
(d) trajectory of X-dot



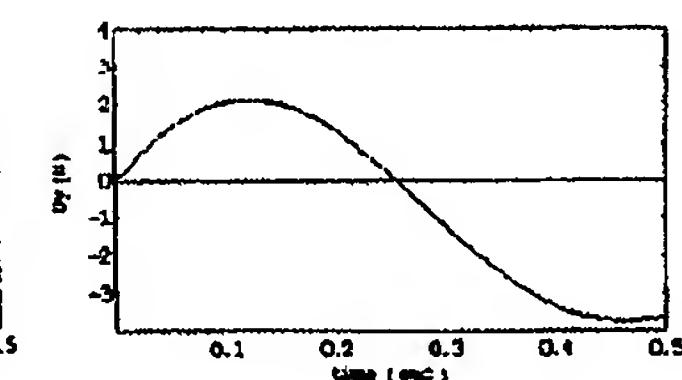
(e) trajectory of Y-dot



(f) trajectory of Z-dot

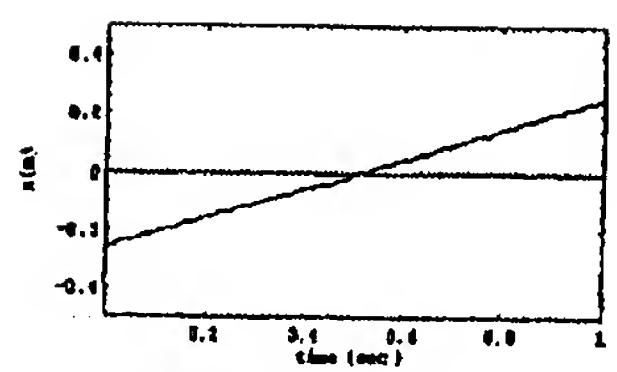
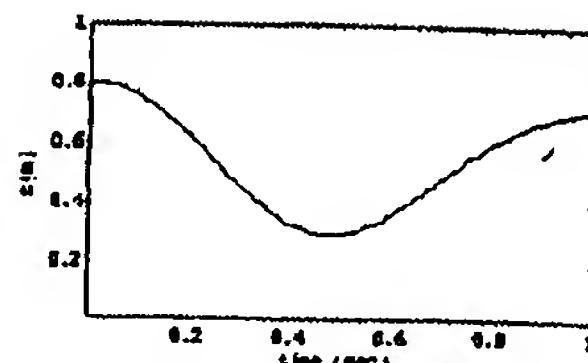
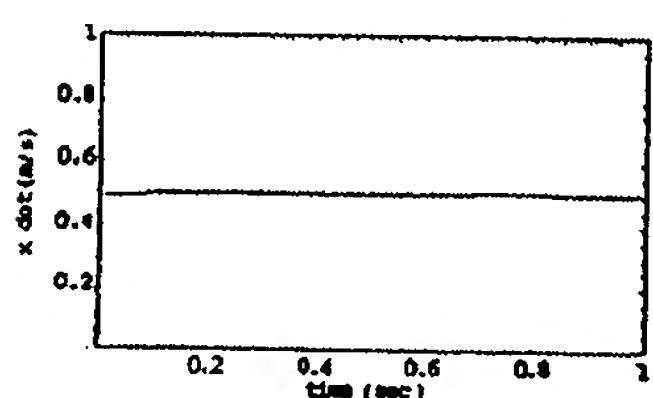
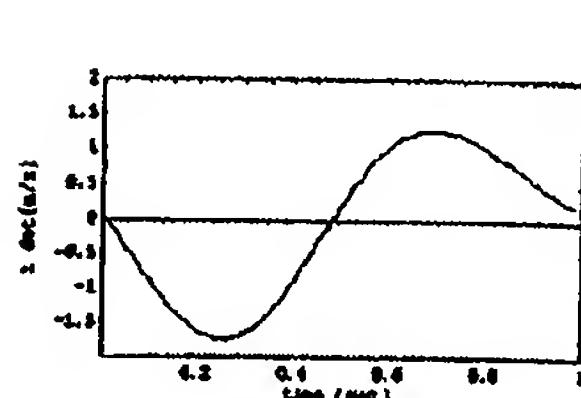
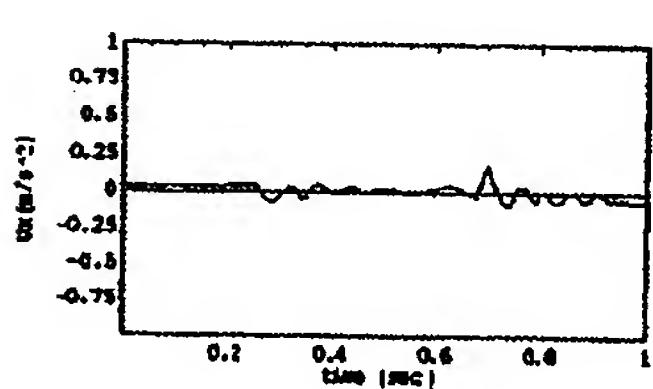
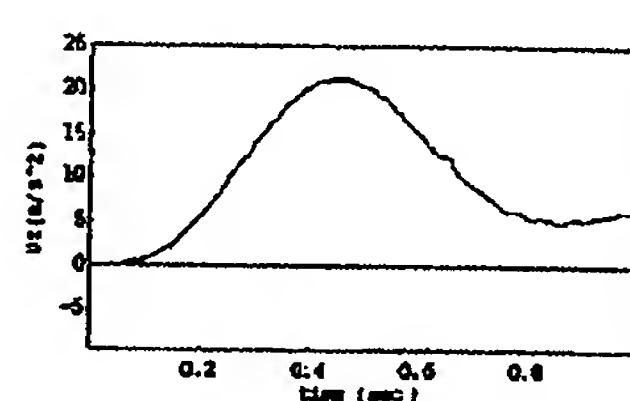
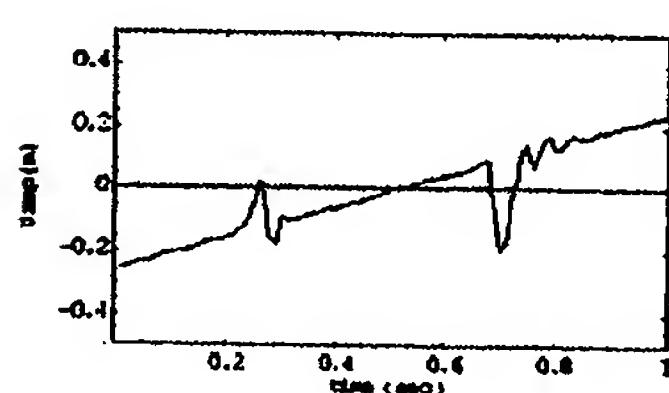


(g) input along X axis

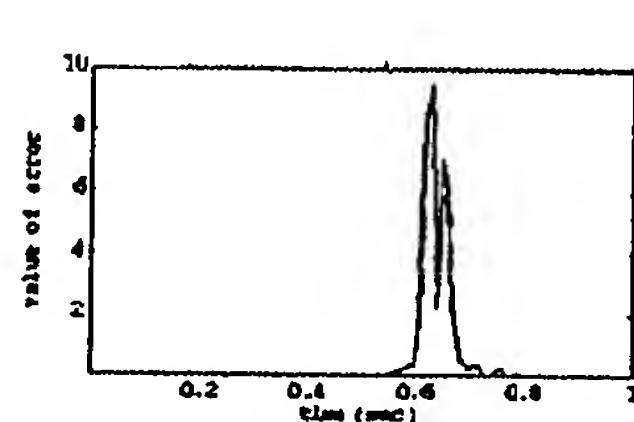
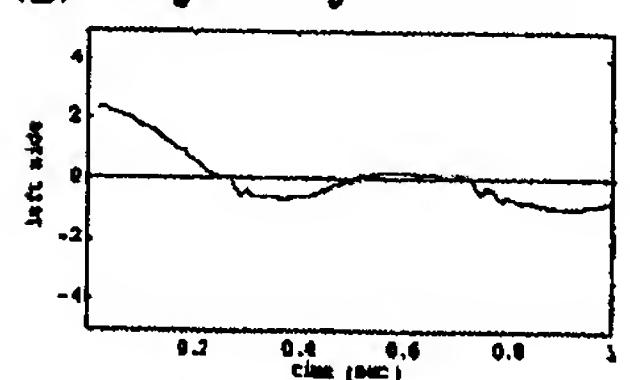


(h) input along Y axis

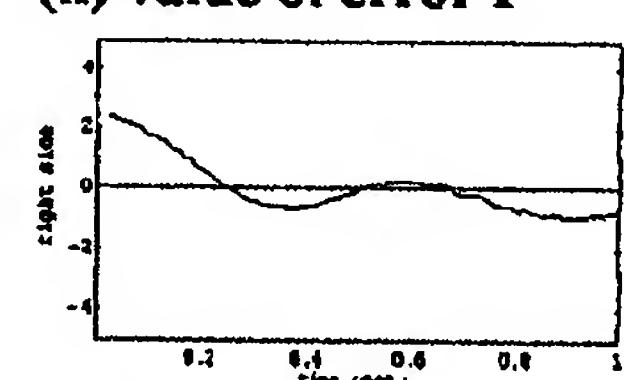
【図4】

(a) trajectory of x (b) position trajectory of z (c) trajectory of \dot{x} (d) trajectory of \dot{z} (e) trajectory of input x (f) trajectory of input z 

(g) trajectory of ZMP

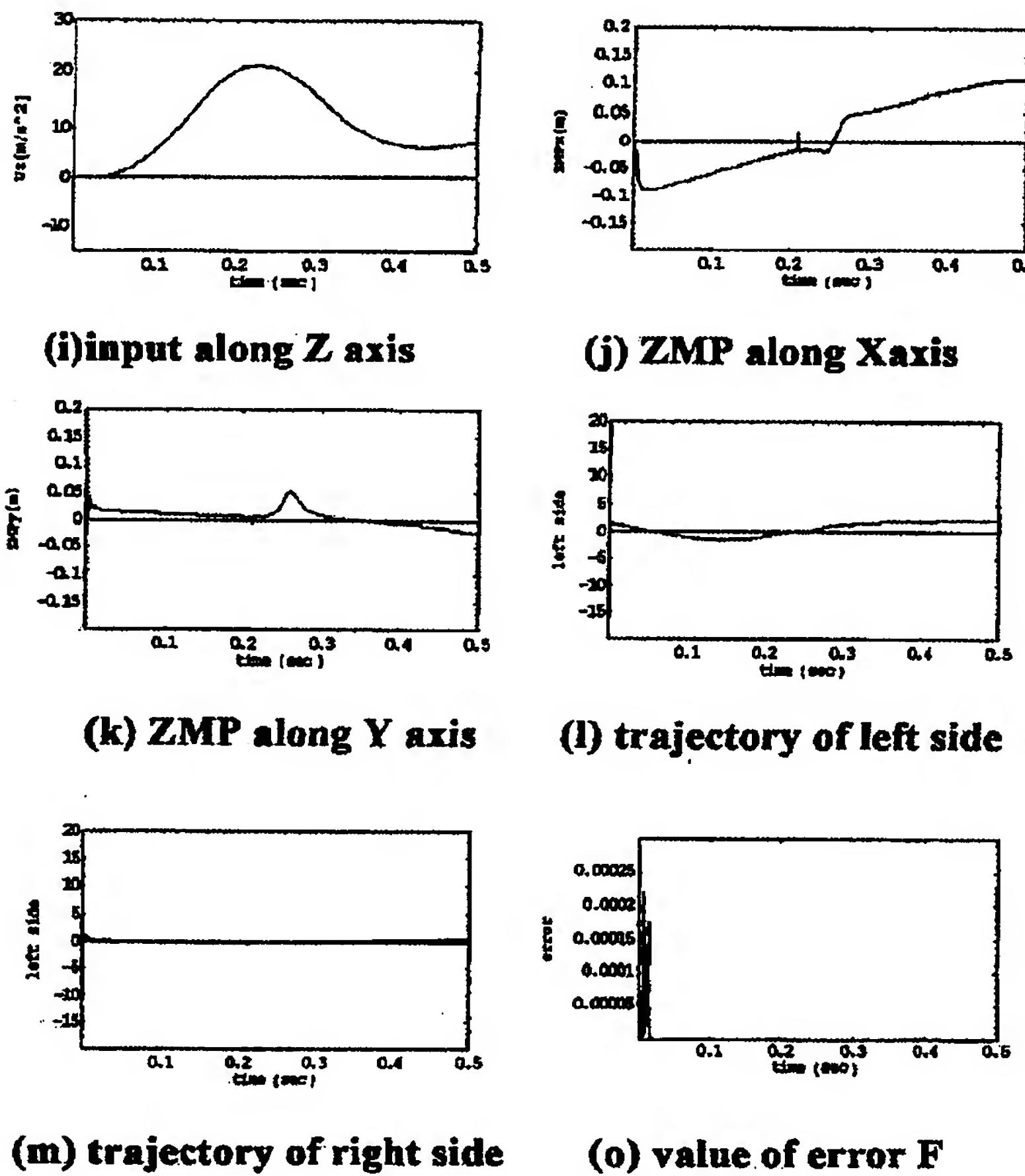
(h) value of error F 

(i) trajectory of left side

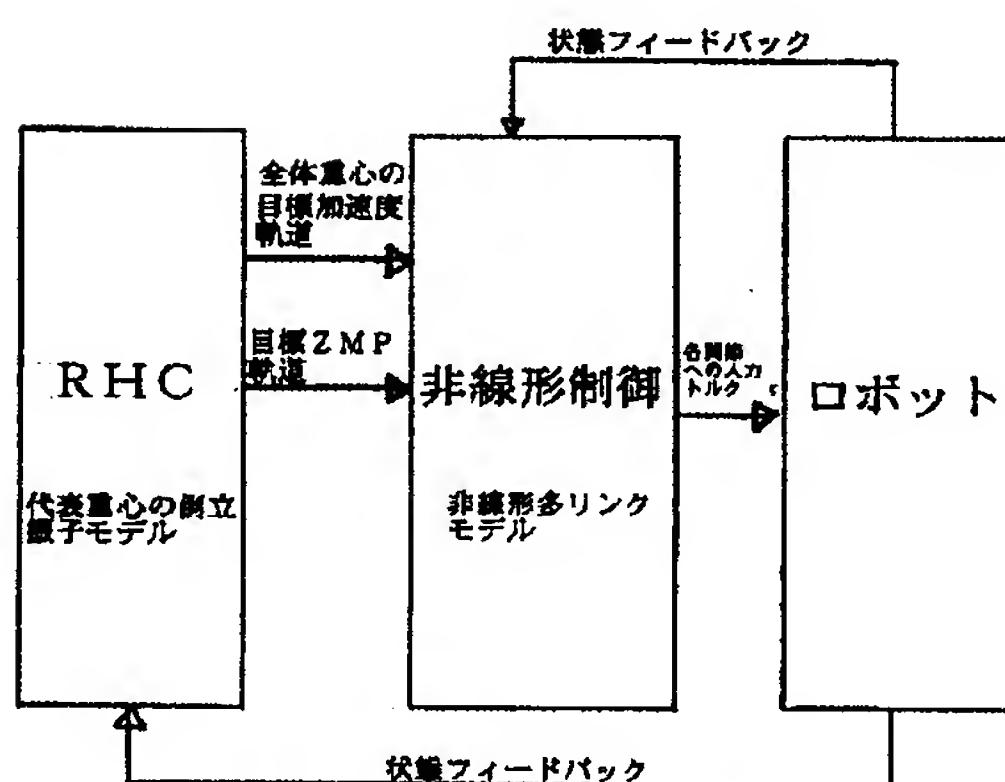


(j) trajectory of right side

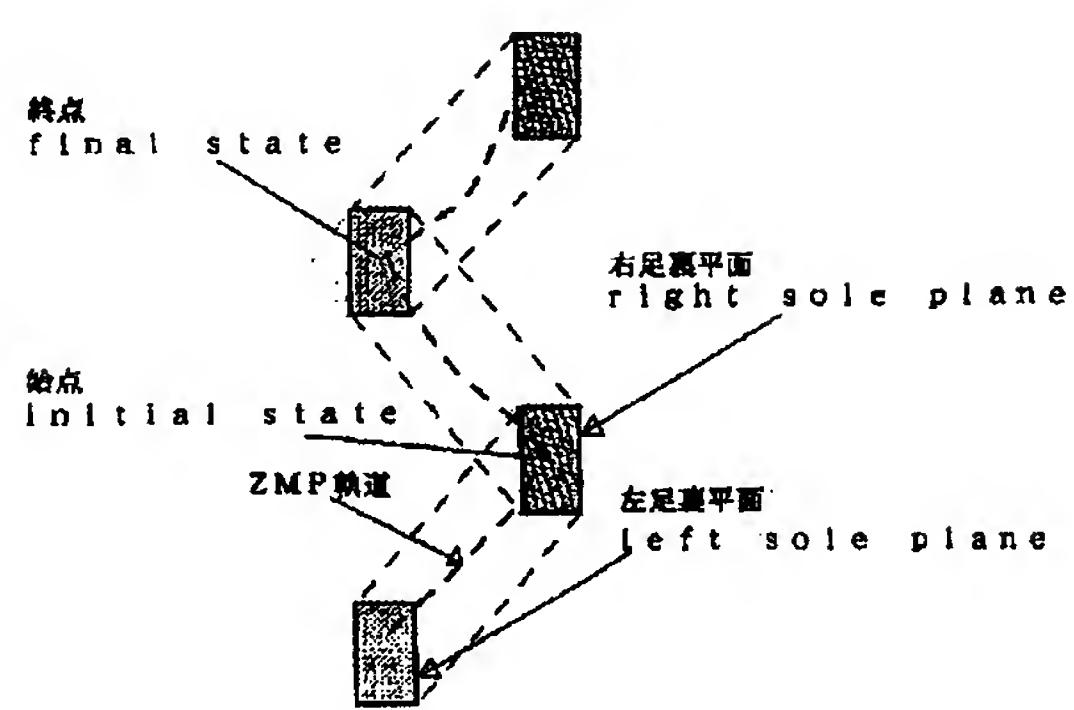
【図6】



【図7】



【図8】



【図9】

